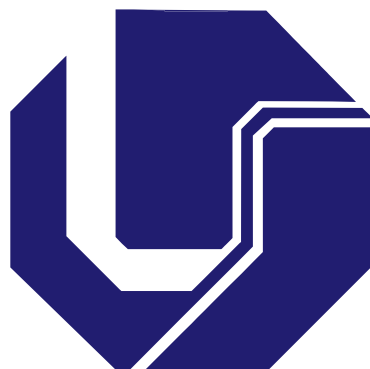


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA BIOMÉDICA



**PLATAFORMA WEB DE REALIDADE AUMENTADA
NÃO-IMERSIVA PARA TRATAMENTO DE
ARACNOFOBIA**

Paulo Camargos Silva

Julho
2019

PAULO CAMARGOS SILVA

**PLATAFORMA WEB DE REALIDADE AUMENTADA
NÃO-IMERSIVA PARA TRATAMENTO DE
ARACNOFOBIA**

Monografia apresentada como parte dos requisitos para a avaliação na disciplina Trabalho de Conclusão de Curso no curso de Graduação em Engenharia Biomédica da Universidade Federal de Uberlândia.

Banca Examinadora:

Prof. Edgard A. Lamounier Jr, PhD – Orientador (UFU)

Profa. Ana Claudia Patrocínio, Dra. – (UFU)

Prof. Leandro Resende Mattioli, MSc. – (CEFET-MG)

Uberlândia

2019

PAULO CAMARGOS SILVA

PLATAFORMA WEB DE REALIDADE AUMENTADA NÃO-IMERSIVA PARA TRATAMENTO DE ARACNOFOBIA

Trabalho aprovado. Uberlândia, 01 de julho de 2019:

Prof. Edgard A. Lamounier Jr, PhD.
Orientador

Profa. Ana Claudia Patrocinio, Dra.
Membro 1

Prof. Leandro Resende Mattioli, MSc.
Membro 2

Uberlândia
2019

Este trabalho é dedicado à minha família, que sem o suporte a mim fornecido, jamais teria conseguido alçar este voo.

Agradecimentos

Os agradecimentos principais são direcionados ao Daniel S. D. Caetano, Gabriel F. Cyrino, Alessandra B. Azar e todos aqueles que contribuíram para o desenvolvimento deste projeto.

Agradecimentos especiais são direcionados ao Grupo de Realidade Virtual e Aumentada (GRVA), ao Laboratório de Engenharia Biomédica (BIOLAB) e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Resumo

A Aracnofobia (fobia de aranhas) é uma das fobias mais comuns, afetando negativamente a qualidade de vida das pessoas. Os tratamentos atuais em sua grande maioria envolvem a exposição do objeto causador do medo ao paciente. A exposição pode ocorrer através de fotografias, desenhos ou *in vivo*.

Embora estas formas de tratamento possuam eficácia comprovada em muitos casos, uma boa parte dos pacientes são desestimulados a iniciarem ou continuarem as sessões devido aos protocolos e abordagem de exposição.

Baseando-se em técnicas de Realidade Aumentada, uma plataforma para tratamento de Aracnofobia foi desenvolvida. Este trabalho descreve esta plataforma construída utilizando ferramentas *Web*. Será descrito a construção do sistema, ferramentas, modo de utilização e as suas funcionalidades.

O sistema tem como principal funcionalidade fornecer um ambiente de exposição dos pacientes ao objeto causador do medo (aranhas) virtuais e o gerenciamento de pacientes e modelos 3D, por parte do terapeuta responsável. Os usuários (paciente e terapeuta) podem realizar acesso através de um computador ou *smartphone* conectados à internet, de qualquer local e horário.

Palavras-chave: Realidade Aumentada; Aracnofobia; Biblioteca de modelos 3D; Modelos 3D; Web.

Abstract

Arachnophobia (spiders phobia) is one of the most common phobias, negatively affecting people's quality of life. Current treatments in the vast majority involve exposure of the object that causes fear to the patient. Exposure can occur through either photographs, drawings or *in vivo*.

Although these methods have proven efficacy in many cases, a large proportion of patients are discouraged from starting or continuing the sessions because of the type of exposition and its protocols.

Based on Augmented Reality techniques, a platform for Arachnophobia treatment was developed. This work describes this web-based platform. The system's construction methodology will be described, as well as the tools, usage and its functionalities.

The main purpose of the system is to provide an environment to expose patients to their virtual object of fear (spiders) and the system's management, by the responsible therapist. Users (patient and therapist) can access the system through a computer or smartphone connected to the internet, from any location and time.

Keywords: 3D Models; Augmented Reality; Arachnophobia; 3D Model Repository; Web.

Lista de Figuras

3.1	Estrutura de uma aplicação SPA.	12
4.1	Relacionamento entre os usuários e o sistema. O sistema é representado pelo <i>front end</i> , <i>back end</i> e <i>database</i>	13
4.2	Casos de Uso do sistema.	16
4.3	Diagrama de sequência para exibição da lista de modelos autorizados à um paciente.	20
4.4	Diagrama de classes das entidades no <i>back end</i>	26
4.5	DER (Diagrama Entidade Relacionamento) do banco de dados.	27
5.1	Tela inicial do perfil do terapeuta.	29
5.2	Tela inicial do perfil do paciente.	30
5.3	Exemplos de modelos 3D que podem ser cadastrados no sistema.	30
5.4	Tela de cadastro de modelos 3D.	31
5.5	Lista de modelos 3D cadastrados.	32
5.6	Tela de edição de informações de modelos 3D.	32
5.7	Exemplo de renderização de modelo 3D na tela do computador.	33

Lista de Abreviaturas e Siglas

API Application Programming Interface

CORS Cross-Origin Resource Sharing

CRP Conselho Regional de Psicologia

CSS Cascading Style Sheets

DOM Document Object Model

DSM-V Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition

glTF GL Transmission Format

GPU Graphical Processing Unit

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

JPA Java Persistence API

JPQL JPA Query Language

JSON JavaScript Object Notation

RA Realidade Aumentada

REST Representational State Transfer

RV Realidade Virtual

SGBD Sistema de Gerenciamento de Banco de Dados

SPA Single-Page Application

SQL Structured Query Language

URL Uniform Resource Locator

XML Extensible Markup Language

WebGL Web Graphics Library

Sumário

1	Introdução	1
1.1	Temática e Motivação	1
1.2	Objetivos	2
1.2.1	Objetivos Gerais	2
1.2.2	Objetivos Específicos	2
2	Estado da arte	3
2.1	Modalidades de Tratamento de Fobias Convencionais	3
2.1.1	Exposição <i>in vivo</i>	3
2.1.2	Exposição Interoceptiva	4
2.1.3	Terapia Cognitiva	4
2.2	Modalidades de Tratamento de Fobias Utilizando RV	4
2.3	Modalidades de Tratamento de Fobias Utilizando RA	5
2.4	Conclusões	5
3	Fundamentação Teórica	6
3.1	Aplicações REST	6
3.2	Visão Geral	7
3.3	Dinâmica da Aplicação REST	7
3.3.1	Anatomia de Uma Requisição (<i>request</i>)	7
3.3.2	Anatomia de Uma Resposta (<i>response</i>)	9
3.4	Ferramentas e Tecnologias	9
3.4.1	<i>Front End</i>	10
3.4.2	<i>Back End</i>	10

3.4.3	Banco de Dados	10
3.4.4	SPA: Single-Page Applications	11
4	Modelagem do Sistema e Arquitetura	13
4.1	Estrutura Principal	13
4.2	Requisitos do Sistema	14
4.2.1	Requisitos Funcionais	14
4.2.2	Requisitos Não Funcionais	15
4.3	Casos de Uso	16
4.3.1	Casos de Uso do Sistema	16
4.4	Diagrama de sequência	19
4.5	<i>Back end</i>	20
4.5.1	Ferramentas e Tecnologias	20
4.5.2	Estrutura	21
4.6	<i>Front End</i>	22
4.6.1	Ferramentas e Tecnologias	22
4.6.2	Estrutura	23
4.6.3	Renderização de Modelos em RA	24
4.7	Banco de Dados	24
4.7.1	Ferramentas e Tecnologias	24
4.7.2	Modelagem	25
5	Resultados	29
5.1	Perfis do Sistema	29
5.2	Cadastro de Modelos 3D	30
5.3	Visualização de Modelos 3D	31
6	Conclusões e Trabalhos Futuros	34
	Bibliografia	36

Introdução

1.1 Temática e Motivação

A Realidade Aumentada (RA) têm sido amplamente utilizada em diversas áreas, desde a área médica (SIELHORST; FEUERSTEIN; NAVAB, 2008) até a educação (BACCA et al., 2014).

Em relação à área médica, a RA tem auxiliado no treinamento de procedimentos (BARSOM; GRAAFLAND; SCHIJVEN, 2016) e em cirurgias (BERNHARDT et al., 2017). O tratamento de distúrbios psicológicos tem sido igualmente beneficiado com a RA (CHICCHI GIGLIOLI et al., 2015).

A RA tem sido também utilizada no tratamento de fobias específicas (JUAN et al., 2005; BOTELLA; BAÑOS et al., 2000). As fobias específicas são caracterizadas como um medo e/ou ansiedade desproporcional, sem justificativa em relação a um objeto e/ou situação (MEDEIROS et al., 2004). O Manual de Diagnóstico e Estatístico de Transtornos Mentais (DSM-V) classifica cinco subtipos de fobias: animais, ambiente natural, sangue-injeção-ferimentos, situacional e outros (APA, 2013).

A Aracnofobia é um tipo de fobia específica envolvendo aranhas (classificada no tipo “animais”) e é um dos tipos mais comuns (BOURDON et al., 1988). Pessoas acometidas deste tipo de fobia desenvolvem uma aversão a qualquer objeto que remeta a aranhas (APA, 2013), sofrendo prejuízo em suas atividades diárias.

Dentre os tratamentos atuais, a exposição *in vivo* é um dos mais eficazes na diminuição dos distúrbios causados pela fobia. Entretanto, muitas vezes os pacientes recusam a se submeterem ao tratamento devido ao medo da exposição (MARKS; O’SULLIVAN, 1992).

Outros fatores desestimulantes para os pacientes seguirem um tratamento são a dificuldade

de deslocamento, de terapeutas e pacientes até locais apropriados e a alocação de recursos (MILOFF et al., 2016). Estes fatores levam a uma alta taxa de evasão dos tratamentos (CHOY; FYER; LIPSITZ, 2007).

Desta forma, a RA pode oferecer uma alternativa a estes problemas. Além disso, a RA tem mostrado resultados semelhantes e com efetividade clínica quando comparada aos métodos de terapia tradicionais (WRZESIEN et al., 2015).

Através da RA, a necessidade de deslocamento é eliminada e a quantidade de recursos pode ser reduzida à um *smartphone* ou computador. Além disso, o paciente pode ter a certeza que durante todo o tratamento não sofrerá nenhum tipo de possível ataque, pois o elemento é virtual. A *web* propicia também um ambiente acessível 24h por dia.

1.2 Objetivos

1.2.1 Objetivos Gerais

Este trabalho tem o objetivo de criar uma plataforma de Realidade Aumentada para tratamento de Aracnofobia. O sistema tem o propósito de ser uma plataforma de RA de baixo custo, fácil utilização e amplo acesso.

1.2.2 Objetivos Específicos

Por ser um sistema baseado na Web, permite-se que o mesmo seja implantado em um servidor conectado a rede mundial de computadores. Assim, o sistema tem o intuito de não possuir restrições rígidas quanto ao *hardware* para sua utilização.

Para que os terapeutas e pacientes possam usufruir da plataforma, é necessário apenas um computador com câmera ou *smartphone* e acesso à internet. Para isto, os usuários devem acessar um *link*, realizar a autenticação e acessar seu perfil. Com isto, a plataforma tem a finalidade de permitir que os usuário possam utilizá-la de qualquer local, não limitando-se aos consultórios dos terapeutas.

Associado a esse sistema, é também apresentado a criação de uma biblioteca de modelos 3D em que terapeutas podem adicionar e editar informações como nível de realismo e categoria dos modelos. Os pacientes, por sua vez, podem visualizar os modelos liberados pelo terapeuta através da plataforma.

Estado da arte

2.1 Modalidades de Tratamento de Fobias Convencionais

Desde a década de 1960, diferentes modalidades de tratamento de fobias vêm sendo empregadas. Como exemplo, pode-se citar a terapia cognitiva, terapia comportamental e outras não tão aceitas como terapia suportiva, hipnoterapia e farmacoterapia (CHOY; FYER; LIPSITZ, 2007).

Na modalidade comportamental, pode-se citar métodos como exposição imaginativa, exposição *in vivo*, exposição interoceptiva, exposição por Realidade Virtual e tensão aplicada (CHOY; FYER; LIPSITZ, 2007). Alguns desses métodos serão descritos a seguir.

2.1.1 Exposição *in vivo*

A exposição *in vivo* envolve a exibição do objeto causador do medo ao paciente. Esta exposição geralmente ocorre de forma gradativa, onde o paciente é exposto, inicialmente a uma situação menos estressante, até uma de maior realismo e estresse. Esta modalidade é, usualmente, dividida em sessões com duração de várias horas.

Este método tem se mostrado efetivo em muitos casos. Entretanto, este tipo tem sido associado a índices de evasão de até 45% (CHOY; FYER; LIPSITZ, 2007). As limitações deste tipo de tratamento envolvem o deslocamento de pacientes e terapeutas até locais apropriados para se realizar a exposição, a alocação de recursos, bem como a dificuldade para replicar situações, resultando na relutância do paciente para se empenhar no tratamento (MISHKIND et al., 2017).

2.1.2 Exposição Interoceptiva

A exposição interoceptiva tem também tem sido utilizada em tratamentos de claustrofobia, tratamentos de ansiedade e ataques de pânico. O método, classificado também como forma de terapia comportamental, demanda que o paciente realize exercícios que trazem sensações físicas semelhantes àsquelas do transtorno, porém em um ambiente controlado.

Este tipo é diferente da exposição *in vivo* onde os sentimentos são provocados pelo exposição direta do objeto causador do medo. A exposição interoceptiva tem sido subutilizado na prática clínica atualmente (BOETTCHER; BRAKE; BARLOW, 2016).

2.1.3 Terapia Cognitiva

A terapia cognitiva possui o foco em reduzir o nível de ansiedade ou medo fazendo com que o paciente reavalie os medos irracionais associados ao objeto causador do medo. Esta metodologia pode ser aplicada de forma independente ou como auxiliar de outro método, por exemplo, com o tratamento com exposição *in vivo*.

Este método tem se mostrado eficaz para o tratamento de claustrofobia, de forma independente e como método auxiliar. Entretanto, a terapia cognitiva não é tão eficaz em fobias envolvendo animais ou medo de voar (CHOY; FYER; LIPSITZ, 2007).

2.2 Modalidades de Tratamento de Fobias Utilizando RV

As técnicas de Realidade Virtual (RV) utilizadas atualmente envolvem a exposição do paciente ao elemento causador do medo de forma artificial, com simulações que podem ser extremamente semelhantes ao real. Sistemas utilizando RV tem se mostrado eficaz em distúrbios psicológicos como estresse pós-traumático, ansiedade e fobias, dores crônicas e vícios (MISHKIND et al., 2017).

Um estudo de metáanálise (MORINA et al., 2015) mostra que a RV possui eficácia significativamente melhor que outros métodos de tratamento, como o de exposição *in vivo*. A RV também tem sido aplicada em tratamentos de fobia de voar (KRIJN et al., 2007), acrofobia (medo de altura), aracnofobia, fobia de tempestades e estresses (MAPLES-KELLER et al., 2017). Estudos de revisão tem mostrado a efetividade deste método (BOTELLA; FERNÁNDEZ-ÁLVAREZ et al., 2017; MAPLES-KELLER et al., 2017).

2.3 Modalidades de Tratamento de Fobias Utilizando RA

A Realidade Aumentada (RA) tem a característica de associar elementos virtuais com elementos reais. Os métodos de tratamentos atuais envolvendo a RA utilizam a exposição do elemento causador do medo ao paciente (elemento virtual) como a RV, porém elementos reais podem ser incluídos durante a exposição.

A RA tem sido utilizada no tratamento de diversas fobias com animais pequenos, como por exemplo baratas (BOTELLA; BRETÓN-LÓPEZ et al., 2010) e aranhas (JUAN et al., 2005). Um estudo comparativo (SUSO-RIBERA et al., 2018) mostrou que metodologias utilizando a RA tem eficácia semelhante às outras técnicas como exposição *in vivo* e RV.

Outro estudo comparativo (BOTELLA; PÉREZ-ARA et al., 2016), realizado com pacientes com fobias de animais pequenos, comparou 31 pacientes que efetuaram o tratamento com exposição *in vivo* com 32 paciente que realizaram o tratamento através de RA. Os pacientes que realizaram o tratamento com RA consideraram o tratamento menos aversivo. O estudo também observou maior aceitação pelo tratamento com RA entre os pacientes .

2.4 Conclusões

Os tratamentos atuais possuem diferentes aplicabilidades e são voltadas para cada tipo de fobia. A técnica de exposição *in vivo* é amplamente utilizada em fobias específicas e tem mostrado resultados eficazes. Entretanto, possui um alto nível de evasão por parte dos pacientes. Razões para isto variam desde relutância à exposição ao elemento causador do medo até questões financeiras.

A alternativa de exposição interoceptiva é utilizada para induzir sensações físicas como sensações de tontura e náuseas. Assim, as fobias geralmente associadas a esta modalidade são claustrofobia e ansiedade.

A técnica de terapia cognitiva é aplicada como um método complementar aos outros métodos, como a exposição *in vivo*. Além disso, a literatura mostra que, esta técnica não é tão eficaz em fobias envolvendo animais.

Os métodos de tratamentos utilizando RV e RA têm obtido resultados semelhantes às técnicas já consolidadas, como a exposição *in vivo*. Entretanto, possui limitações, como exemplo, a dificuldade em reproduzir um ambiente fiel ao real. Outro ponto a ser melhorado diz respeito a usabilidade do sistema. Usuários podem relatar sensações de tontura ao utilizar as soluções.

Fundamentação Teórica

3.1 Aplicações REST

A arquitetura REST (*Representational State Transfer*) foi desenvolvida pelo cientista da computação norte-americano Roy Fielding e apresentada pela primeira vez em sua tese de doutorado, no ano de 2000. Ela surgiu da necessidade de interoperabilidade entre aplicações servidas sobre o protocolo HTTP (*HyperText Transfer Protocol*) (FIELDING; TAYLOR, 2000).

Aplicações REST são habitualmente chamadas de *web services* e por vezes são referidas como RESTful. A arquitetura REST é definido por 6 restrições (*constraints*) que servem como guia. São elas:

- *Uniform interface* (interface uniforme)
- *Client-server* (cliente-servidor)
- *Stateless* (sem estados)
- *Cacheable* (cache)
- *Layered system* (sistema em camadas)
- *Code on demand* (opcional) (código sob demanda)

Estas retrições, além de servirem como um guia, limitam a maneira como o sistema deve funcionar seguindo padrões, permitindo a escalabilidade, interoperabilidade, simplicidade, performance, visibilidade, portabilidade e confiabilidade.

3.2 Visão Geral

A construção de uma aplicação REST permite que as partes do sistema (aplicações) sejam desenvolvidas de forma separada. Assim, o sistema foi desenvolvido em duas etapas (camadas) principais: desenvolvimento do *front end* e *back end*.

O termo *front end* é utilizado para se referir à camada de apresentação de uma determinada aplicação. Esta camada é a interface entre usuário e *back end*. É através dela que o usuário interage de forma direta com um sistema. O *front end* é responsável pela apresentação de informações gerais ao usuário, instruções de utilização, formulários para preenchimento, entre outros.

Esta camada é entregue ao cliente (usuário final) por um servidor e processada pelo navegadores (*browsers*), na máquina do cliente. As três principais ferramentas utilizadas na construção da camada de *front end* são o HTML, CSS e *JavaScript*. A combinação das três ferramentas gera a camada visual do sistema.

O termo *back end* é utilizado se refere à camada responsável pela lógica de funcionamento de um sistema, acesso e manipulação do banco de dados e exposição de dados. Seu funcionamento ocorre em um servidor remoto ou local, conectado à *internet*.

3.3 Dinâmica da Aplicação REST

O funcionamento de uma aplicação REST é baseado em requisições (*requests*) e respostas (*responses*) realizadas à uma API (Application Programming Interface). As requisições são efetuadas através de URLs (Uniform Resource Locator). Após uma requisição feita ao servidor e seu respectivo processamento, uma resposta é retornada ao cliente que efetuou a requisição.

3.3.1 Anatomia de Uma Requisição (*request*)

As requisições feitas pelo cliente (solicitante) seguem uma estrutura padronizada. Esta estrutura é composta de até quatro componentes:

- Rota (*endpoint/route*)
- Método (*method*)
- Cabeçalhos (*headers*)

- Corpo (*body/message*)

A rota diz respeito à URL em que a requisição será feita. Ela é composta pela raiz, que é o início da rota (*root-endpoint*) e o caminho (*path*) que determina qual recurso (*resource*) será utilizado do API. A seguir, um exemplo de uma rota completa:

```
http://localhost:8080/api/user/1
```

A arquitetura REST convencionou a utilização correta dos métodos HTTP nas aplicações. O método (*method*) determina o tipo da requisição a ser efetuada. Os cinco principais métodos HTTP são:

- *GET*
- *POST*
- *PUT*
- *PATCH*
- *DELETE*

O método *GET* é utilizado para solicitar informações ao servidor. O método *POST* é utilizado para envio de informação nova ao servidor, como por exemplo, num cadastro de usuário. O método *PUT* é utilizado para enviar informações ao servidor, substituindo uma outra já existente de forma completa (atualização completa).

O método *PATCH* é utilizado para enviar informação ao servidor substituindo parcialmente dados num objeto já existente (atualização parcial). O método *DELETE* é usado para solicitar a remoção de um registro existente.

Os cabeçalhos (*headers*) permitem ao cliente e servidor passar informações extras. Cada cabeçalho HTTP consiste de pares chave-valor separados por dois pontos. Um exemplo de informação presente no cabeçalho de requisições é a autorização de acesso a um recurso da API, contendo uma chave encriptada (*token*). Exemplo:

```
{  
  "Authorization": "p2WEM5oaon5o35xWapos...oi1o20psodfbGvVL0w6"  
}
```

O corpo (*body/message*) contém a informação que deve ser enviada ao servidor. As informações no corpo de uma requisição são geralmente representadas em formato *JSON* (JavaScript Object Notation), porém o formato *XML* (Extensible Markup Language) também pode ser utilizado.

O corpo de uma requisição só é utilizado nos métodos do tipo *POST*, *PUT*, *PATCH* e *DELETE*. Os dados são passados em forma de chave-valor. A seguir, um exemplo utilizado em um método *POST*:

```
1 {  
2     "nome": "João da Silva",  
3     "idade": 27,  
4     "email": "joao@exemplo.com"  
5 }
```

3.3.2 Anatomia de Uma Resposta (*response*)

Assim como as requisições, as respostas também possuem uma anatomia própria. Uma simples resposta contém os seguintes componentes:

- Código *status* de resposta HTTP (HTTP *Status Code*)
- Cabeçalho (*Headers*)
- Corpo (opcional) (*body/message*)

O código de *status* de resposta HTTP indica a situação de uma requisição feita ao servidor, isto é, se foi corretamente concluída ou não. Podem ser agrupados em respostas informativas ou de sucesso, redirecionamentos, erros de cliente ou de servidor. Por exemplo, o código “200 OK” significa que uma requisição foi recebida e processada com sucesso.

3.4 Ferramentas e Tecnologias

A seguir, são apresentadas as ferramentas e tecnologias utilizadas na construção do sistema proposto neste trabalho.

3.4.1 *Front End*

As principais ferramentas utilizadas no desenvolvimento da aplicação de *front end* foram a linguagem de marcação de texto HTML (Hypertext Markup Language), a ferramenta de estilização de textos CSS (*Cascading Style Sheets*) e a linguagem de programação *JavaScript* com o *framework* *Vue.js*.

O *Vue.js* permite o desenvolvimento de aplicações SPA (*Single Page Application*), descritas posteriormente no presente capítulo, e fornece suporte a ferramentas externas ao seu ecossistema.

Estas ferramentas em conjunto, permitem a apresentação de dados, a estilização e sua manipulação. Além das ferramentas descritas acima, foram utilizados *frameworks* para tarefas específicas, como por exemplo, um cliente HTTP para requisições. A apresentação de modelos em RA foi feita utilizando o *framework* A-Frame em conjunto com o *framework* *AR.js*.

3.4.2 *Back End*

O desenvolvimento do *back end* foi realizado utilizando a linguagem de programação *Java* (v. 11) com o *framework* *Spring*¹. Este *framework* fornece ferramentas úteis para controle de aplicações REST, bem como mecanismos de acesso aos dados do banco de dados utilizando mapeamento e mecanismos de segurança.

3.4.3 Banco de Dados

Aplicações frequentemente necessitam de algum sistema de armazenamento de informação. A camada de dados é construída, muitas vezes, em cima de um banco de dados. O banco de dados é um agrupamento de dados (informações) que se relacionam entre si sobre um domínio (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016). A definição dos mecanismos de armazenamento e de manipulação dos dados está diretamente ligado ao gerenciamento do banco de dados.

Para permitir a interação do usuário com os dados apenas, “escondendo” dos usuários os detalhes de uma implementação e funcionamento de banco de dados, os sistemas são representados por três camadas abstratas principais:

- Camada física

¹ Documentação disponível em <https://spring.io/>

- Camada lógica
- Camada de visualização

A camada física é aquela de mais baixo nível (mais próxima da máquina). Representa a maneira como os dados são armazenados. A camada lógica é a intermediária e representa quais dados são armazenados e seus relacionamentos.

A camada de visualização é aquela que possui o nível mais alto (mais próximo do usuário) e representa a visualização de dados mais importantes para o usuário, que pode ser apenas parte de toda a informação contida no banco de dados (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016).

Os modelos de dados (*data models*) são uma coleção de técnicas conceituais para a criação (*design*) de um banco de dados. As quatro principais categorias são:

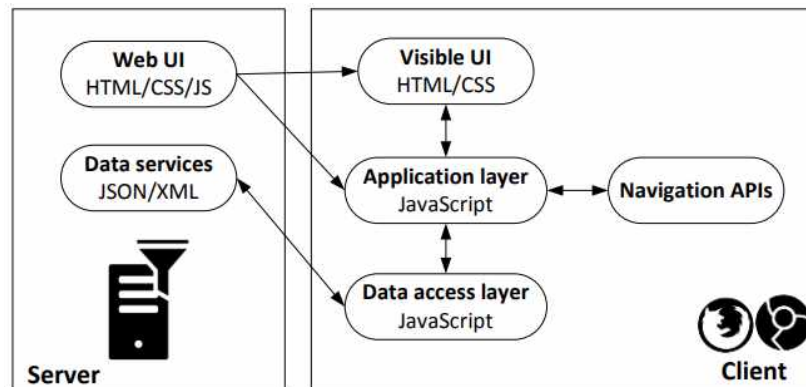
- Modelo relacional
- Modelo entidade-relacionamento (ER)
- Modelo baseado em objetos
- Modelo de dados semiestruturado

O modelo relacional é representado por uma coleção de tabelas, as relações entre si e seus tipos de dado. O modelo entidade-relacionamento (ER) é representado por uma coleção básica de objetos (entidades). O modelo baseado em objetos pode ser visto como uma associação do modelo ER com conceitos de Programação Orientada a Objetos.

3.4.4 SPA: Single-Page Applications

SPA (*Single-Page Applications* ou Aplicações de Página Única) são aplicações Web em que o todo o código HTML, CSS e *JavaScript* é obtido apenas em um único carregamento. Este tipo de aplicação permite ao usuário uma experiência contínua, sem interrupções de carregamento da página, semelhante a aplicações *desktop*. A estrutura geral de uma aplicação deste tipo pode ser observada na Figura 3.1.

Figura 3.1: Estrutura de uma aplicação SPA.



Fonte: (JOSEPH, 2015).

Em aplicações SPA, a interação pode ocorrer com um ou mais servidores. Estas interações ocorrem através de requisições que atendam o padrão REST e são realizadas de maneira fluída, isto é, são efetuadas a medida que o usuário interage com a aplicação e de modo assíncrono.

As aplicações deste tipo são construídas utilizando componentes individuais que podem ser substituídos e/ou atualizados de forma independente (MESBAH; VAN DEURSEN, 2007). Estas alterações, entre componentes, ocorrem pela interação do usuário com a aplicação ou do recebimento e envio de dados a um servidor.

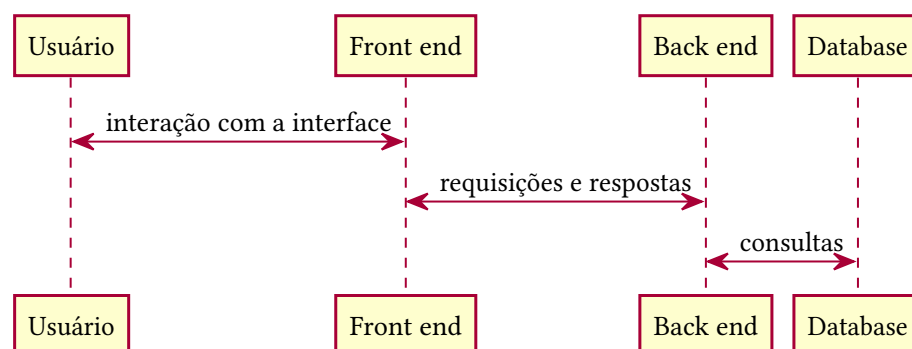
Modelagem do Sistema e Arquitetura

Neste capítulo, é explicada a modelagem do sistema proposto e sua arquitetura. São detalhados nesta seção os requisitos do sistema, casos de uso e ferramentas utilizadas na construção do trabalho.

4.1 Estrutura Principal

O funcionamento do sistema é baseado em requisições da aplicação de *front end* para a aplicação de *back end*. O *back end*, por sua vez, é responsável pela lógica de negócios da aplicação e consultas no banco de dados. A Figura 4.1 exibe o relacionamento entre os usuários e o sistema.

Figura 4.1: Relacionamento entre os usuários e o sistema. O sistema é representado pelo *front end*, *back end* e *database*.



Fonte: O autor (2019).

4.2 Requisitos do Sistema

4.2.1 Requisitos Funcionais

Os requisitos funcionais de um sistema referem-se à maneira como um sistema deve se comportar, mais especificamente, como um sistema deve funcionar. Abaixo, são definidos os requisitos para o funcionamento da plataforma desenvolvida.

RF_01 Cadastro e edição de dados de pacientes

Descrição O sistema deve permitir que os usuários realizem alterações em dados pessoais quando pertinentes.

Entrada Dados pessoais simples e de credenciamento como nome de usuário.

Processo O usuário deverá preencher um formulário e enviar as alterações nos dados.

RF_02 Cadastro e edição de dados de modelos 3D

Descrição O sistema deve permitir que os usuários (terapeutas) realizem o cadastro e edição de informações de modelos 3D de aranhas.

Entrada Dados do modelo como nível e nome do modelo.

Processo O usuário deverá preencher um formulário e enviar as alterações nos dados.

RF_03 Visualização dos modelos 3D

Descrição O sistema deve permitir que os usuários visualizem os modelos cadastrados.

Entrada Nome do modelo a ser visualizado.

Processo O usuário deverá buscar pelo modelo e, ao clicar sobre o mesmo, poderá visualizá-lo.

RF_04 Associar categoria (nível de realismo) aos modelos 3D

Descrição O sistema deve permitir que terapeutas categorize os modelos 3D cadastrados por níveis de realismo.

Entrada Categorias: iniciante, intermediário e avançado.

Processo Ao cadastrar ou editar informações do modelo, o terapeuta deverá selecionar uma das categorias listadas.

RF_05 Associar um nível de evolução ao paciente

Descrição O sistema deve permitir que terapeutas associe um nível de evolução ao paciente cadastrado.

Entrada Categorias: iniciante, intermediário e avançado.

Processo Ao cadastrar ou editar um paciente, o terapeuta deverá selecionar uma das categorias listadas.

4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais de um sistema referem-se a imposições e/ou restrições sobre o funcionamento de um sistema. A seguir, são listados os requisitos não funcionais do sistema.

RNF_01 O cadastro de pacientes novos deverá ser feito apenas pelo terapeuta responsável.

RNF_02 Alteração de nível de paciente somente poderá ser feita pelo terapeuta responsável.

RNF_03 Somente usuários credenciados (com nome de usuário e senha) poderão acessar o sistema.

RNF_04 Os pacientes poderão visualizar somente os modelos 3D abaixo de seu nível e que seu terapeuta responsável cadastrou.

RNF_05 A visualização dos modelos 3D deverá ser feita através de exibição de marcador para câmera, na tela do computador.

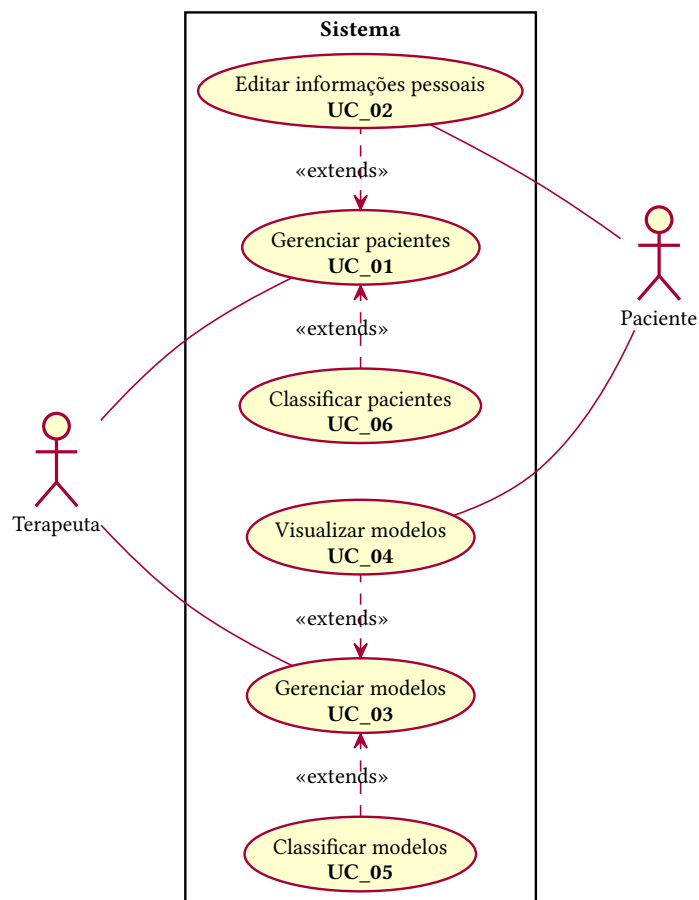
4.3 Casos de Uso

O diagrama de Casos de Uso é uma forma de demonstrar os requisitos funcionais de um sistema, descrevendo as interações entre os usuários e o próprio sistema (FOWLER; KOBRYN; SCOTT, 2004). Ele é composto de diferentes cenários possíveis e os passos possíveis que o usuário pode tomar, para realizar uma ação no sistema.

4.3.1 Casos de Uso do Sistema

O sistema permite a interação dos paciente e terapeutas, de forma separada, em perfis. Os perfis auxiliam na separação e controle de atribuições de cada usuário específico. O diagrama de Casos de Uso da Figura 4.2 exibe as formas de interação possíveis dos usuários com perfil de Terapeuta e Paciente.

Figura 4.2: Casos de Uso do sistema.



Fonte: O autor (2019).

UC_01 - Gerenciar pacientes

Requisitos Associados

- RF_01

Atores Terapeuta.

Resumo O sistema permitirá ao Terapeuta o cadastro, edição e exclusão de pacientes no sistema.

Cenário Principal

Cadastro:

1. O Terapeuta cadastra o usuário, definindo nome de usuário, email e senha para o paciente.
2. O usuário acessa seu perfil com nome de usuário e senha definidos pelo Terapeuta.
3. O Paciente deverá atualizar seu nome de usuário e senha.

UC_02 - Editar informações pessoais

Requisitos Associados

- RF_03
- RF_04

Atores Terapeuta e Paciente.

Resumo O sistema permitirá ao Terapeuta e Paciente a visualização dos modelos 3D cadastrados.

Cenário Principal Visualização do modelo pelo Terapeuta:

1. O usuário acessa a tela de modelos.
2. O usuário abre o modelo 3D desejado.
3. O usuário seleciona a opção “Visualizar modelo 3D”

Cenário alternativo Visualização do modelo pelo Paciente: No caso do paciente, os modelos listados serão aqueles que seu terapeuta responsável cadastrou abaixo do seu nível de realismo atual do tratamento.

UC_03 - Gerenciar modelos

Requisitos Associados

- RF_02
- RF_03

Atores Terapeuta.

Resumo O sistema permitirá ao Terapeuta o cadastro, edição e exclusão de modelos 3D no sistema.

Cenário Principal Cadastro:

1. O Terapeuta acessa a tela de cadastro de modelos.
2. O Terapeuta preenche informações do modelo como nível e nome no formulário.
3. O Terapeuta seleciona o arquivo do modelo em seu computador.
4. O Terapeuta submete o formulário.

UC_04 - Visualizar modelos

Requisitos Associados

- RF_02
- RF_03

Atores Terapeuta.

Resumo O sistema permitirá ao Terapeuta o cadastro, edição e exclusão de modelos 3D no sistema.

Cenário Principal Cadastro:

1. O Terapeuta acessa a tela de cadastro de modelos.
2. O Terapeuta preenche informações do modelo como nível e nome no formulário.
3. O Terapeuta seleciona o arquivo do modelo em seu computador.
4. O Terapeuta submete o formulário.

UC_05 - Classificar modelos 3D

Requisitos Associados

- RF_04

Atores Terapeuta.

Resumo O sistema permitirá o terapeuta a associação de um nível de realismo aos modelos 3D: iniciante, intermediário e avançado.

UC_06 - Classificar pacientes por evolução

Requisitos Associados

- RF_05

Atores Terapeuta.

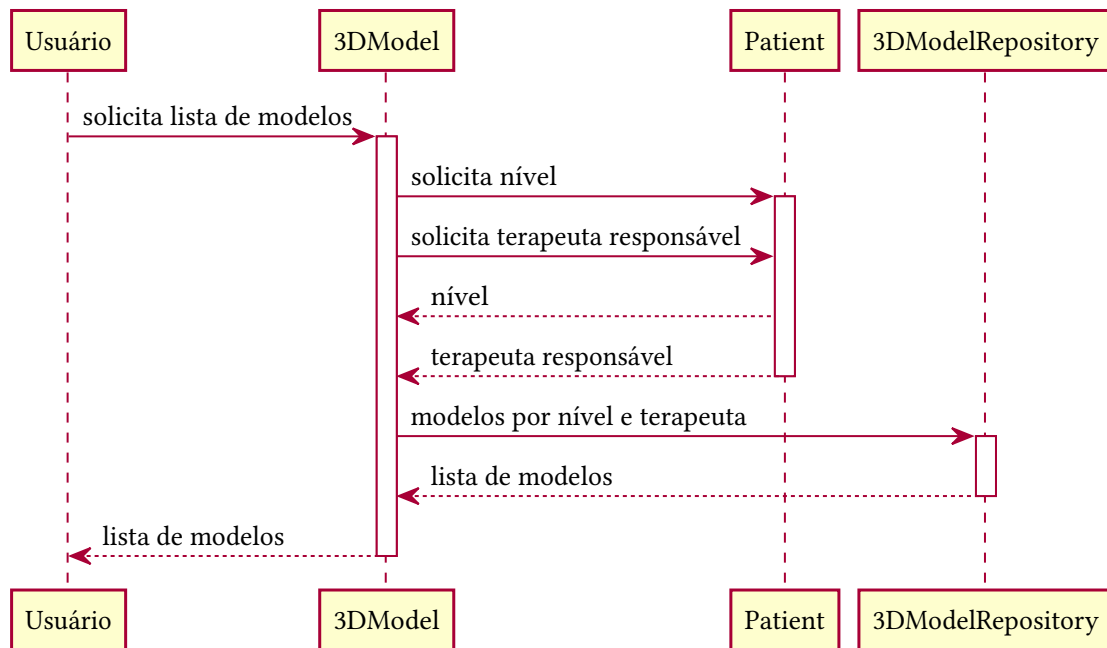
Resumo O sistema permitirá o terapeuta a associação de um paciente a um dos níveis de evolução: iniciante, intermediário e avançado.

4.4 Diagrama de sequência

O diagrama de sequência permite entender como um grupo de objetos se comporta e interage (FOWLER; KOBRYN; SCOTT, 2004). Para melhor entendimento de como o paciente irá interagir com os modelos 3D, foi definido e implementado uma sequência de passos até a exibição do modelo.

Dessa forma, garante-se que, os modelos 3D exibidos a um paciente sejam somente aqueles cujo nível de realismo esteja abaixo da sua classificação (inciante, intermediário e avançado). A Figura 4.3 exibe a sequência de passos no sistema, desde a solicitação da lista de modelos 3D até o seu retorno.

Figura 4.3: Diagrama de sequência para exibição da lista de modelos autorizados à um paciente.



Fonte: O autor (2019).

4.5 *Back end*

4.5.1 Ferramentas e Tecnologias

A principal linguagem de programação utilizada no *back end* foi o Java, na versão 11. Para implementação do *web service* responsável por tratar as requisições e respostas, foi utilizado o *framework* Spring 5.

O *framework* Spring é composto de diversos módulos e fornece diversas ferramentas para o desenvolvimento de *web services*. Abaixo, alguns exemplos:

- *Spring Core Container*: a base do *framework*;
- *AOP (Aspect-oriented Programming)*: suporte a Programação Orientada a Aspecto através de anotações;

- *Data Access*: ferramenta de auxílio ao acesso a dados relacionais e não-relacionais de banco de dados;
- *Authentication e authorization*: ferramentas auxílio a autenticação e autorização, com protocolos de segurança já implementados.

O *Spring Boot* é um pacote contendo vários módulos do *Spring* prontos para uso. Neste pacote, todas as dependências do projeto (módulos de terceiros) são agrupadas através do gerenciador de projetos *Maven*¹. As dependências do projeto são gerenciadas através de um arquivo de configuração chamado *pom.xml*.

As dependências utilizadas neste projeto foram:

- *Web*: fornece as ferramentas básicas de requisição e resposta para aplicações REST;
- *Security*: fornece as ferramentas de autorização e autenticação;
- *JPA*: fornece ferramentas como ORM (Object Relational Mapping), facilitando as operações de acesso ao banco de dados;
- *MySQL JDBC Driver*: dependência de conexão da aplicação em Java com o MySQL;
- *Lombok*: gerador de código em arquivos *.class* como *getters*, *setters* e construtores de classes;

A dependência *Web* fornece as ferramentas básicas de requisição e resposta para aplicações REST.

4.5.2 Estrutura

A aplicação foi construída utilizando a arquitetura REST. Para isto, o projeto de *back end* teve sua estrutura dividida em vários pacotes, sendo eles:

- *config*: pacote de configuração, contendo classes com configurações como armazenamento de arquivo, *CORS* (Cross-Origin Resource Sharing), proteção de rotas (*routes/endpoint*) e outros mecanismos de segurança;
- *controller*: pacote contendo as classes responsáveis pelo recebimento de requisições e envio de respostas. As rotas do *back end* são definidas neste pacote;

¹Documentação disponível em <https://maven.apache.org/>

- *entity*: pacote contendo as entidades (modelos) do banco de dados. Neste pacote, as classes são mapeadas de forma a corresponderem com o banco de dados;
- *exception*: pacote contendo as exceções personalizadas. Estas exceções são lançadas e retornadas como respostas no caso de algum erro do programa;
- *payload*: pacote contendo as classes representando as requisições e respostas;
- *repository*: pacote contendo interfaces que estendem a classe *JpaRepository*, responsável por métodos que realizam consultas no banco de dados;
- *security*: contém classes relacionadas a segurança do projeto, como geração do *token JWT*;
- *service*: pacote contendo as classes de serviço que fazem o intermédio entre os *controllers* e repositórios;

4.6 Front End

4.6.1 Ferramentas e Tecnologias

A camada de *front end* contou com a utilização de várias ferramentas e *frameworks*. A principal linguagem de programação utilizada nesta etapa foi o JavaScript, com o *framework* VueJS. Além disso, foram utilizados outros módulos externos, como o módulo *axios* para requisições HTTP. O HTML e as folhas de estilo CSS também foram utilizados para a estilização das páginas.

Os módulos e bibliotecas utilizadas no *front end* foram:

- *axios*²: cliente HTTP para navegadores. Utilizado na realização de requisições e recebimento de respostas a qualquer servidor REST, com suporte a *JSON*;
- *vue-recaptcha*³: componente Google ReCaptcha⁴ para o VueJS. Este componente de segurança é utilizado em formulários para detecção e proteção contra preenchimentos realizados por robôs (*bots*);

²Documentação disponível em <https://github.com/axios/axios>

³Documentação disponível em <https://www.npmjs.com/package/vue-recaptcha>

⁴Documentação disponível em <https://www.google.com/recaptcha/intro/v3.html#>

- *vue-router*⁵: biblioteca oficial do *framework* VueJS para definição de rotas na aplicação *front end*;
- *vuex*⁶: biblioteca para gerenciamento de estados. Com esta biblioteca é possível organizar os dados da aplicação de *front end* de forma centralizada, facilitando o compartilhamento entre os componentes e reduzindo o número de requisições ao *back end*;
- *vueify*: biblioteca com componentes visuais, como listas, botões e formulários, estilizados no padrão *Material Design*, criado pelo Google;

4.6.2 Estrutura

A linguagem JavaScript permite a manipulação direta dos elementos do DOM (*Document Object Model*) em um navegador. Isto possibilitou a criação de páginas utilizando o conceito SPA. A manipulação do DOM ocorre através da substituição de componentes.

O desenvolvimento da aplicação de *front end* foi feito através de componentes. O componente é a estrutura mínima de uma página SPA. São arquivos com a extensão *.vue* contendo código HTML, CSS e JavaScript. Esta maneira de construção permite a reutilização de componentes em diversos lugares da aplicação de *front end*.

A aplicação de *front end* gira em torno de quatro pacotes principais, cada um contendo um ou mais componentes. Os pacotes principais são:

- *therapists*: pacote contendo os componentes relacionados ao terapeuta, como por exemplo, o formulário para cadastro de terapeuta;
- *patients*: pacote contendo os componentes relacionados aos pacientes. Neste pacote se encontram componentes como listagem e cadastro (formulário) de pacientes;
- *models*: pacote contendo os componentes relacionados aos modelos 3D. Neste pacote se encontram componentes como listagem, cadastro e renderização de modelos 3D;
- *miscellaneous*: pacote contendo componentes diversos, amplamente reutilizados na aplicação de *front end*. Como exemplo, neste pacote se encontram componentes para exibição de mensagens de erro, carregamento de dados e barras de navegação da página;

⁵Documentação disponível em <https://router.vuejs.org/>

⁶Documentação disponível em <https://router.vuejs.org/>

4.6.3 Renderização de Modelos em RA

A exibição dos modelos 3D para o usuário é feita com base em dois *frameworks* principais: *A-Frame*⁷ e *AR.js*⁸. O primeiro, permite a criação de aplicações que utilizam Realidade Virtual. O segundo utiliza o *A-Frame* como base e permite a criação de aplicações que utilizam Realidade Aumentada. A associação entre os dois *frameworks* permite a criação de aplicações utilizando poucas linhas de código.

Ambos *frameworks* foram construídos com base em um outro *framework*, o *Three.js*⁹. Esta biblioteca utiliza o WebGL (*Web Graphics Library*) e permite a criação e exibições de animações 3D utilizando a linguagem *JavaScript* aceleradas pela GPU.

O framework *Three.js* e *A-Frame* permite a utilização de arquivos de modelos 3D de diversas extensões, como *.fbx*, *.dae* (*collada*), *.json*, *.glTF*, entre outros. Neste trabalho, foi escolhido o formato *.glTF* devido aos benefícios que o mesmo traz, como tamanho final reduzido, compactação de modelo e texturas e por ser recomendado pelos *frameworks*.

4.7 Banco de Dados

4.7.1 Ferramentas e Tecnologias

A camada da aplicação de *back end* possui contato direto com a camada de dados. Foram definidas diferentes tabelas e relacionamentos entre as entidades. As principais ferramentas utilizadas no desenvolvimento e implantação do sistema foram:

- MySQL¹⁰: sistema de gerenciamento de banco de dados (SGBD) relacional;
- SQL¹¹: linguagem utilizada na manipulação de dados em um SGBD, sendo a variante utilizada advinda do MySQL;
- JPQL¹²: linguagem baseada em SQL utilizada dentro do ecossistema do Java para criação de *queries* personalizadas;
- *phpMyAdmin*¹³: software livre auxiliar para administração do MySQL;

⁷Documentação disponível em <https://aframe.io/>

⁸Documentação disponível em <https://github.com/jeromeetienne/AR.js>

⁹Documentação disponível em <https://threejs.org/>

¹⁰Documentação encontrada em <https://dev.mysql.com/>

¹¹Documentação encontrada em <https://dev.mysql.com/doc/refman/8.0/en/sql-syntax.html>

¹²Documentação encontrada em <https://www.objectdb.com/java/jpa/query>

¹³Documentação disponível em <https://www.phpmyadmin.net/>

4.7.2 Modelagem

A modelagem foi realizada baseada no *design* das classes no *back end*(secção 3.4.3). Para isto, as entidades necessárias para a aplicação foram definidas e implementadas a partir de classes (*entities*) utilizando o código em *Java*. O diagrama pode ser observado na Figura 4.4.

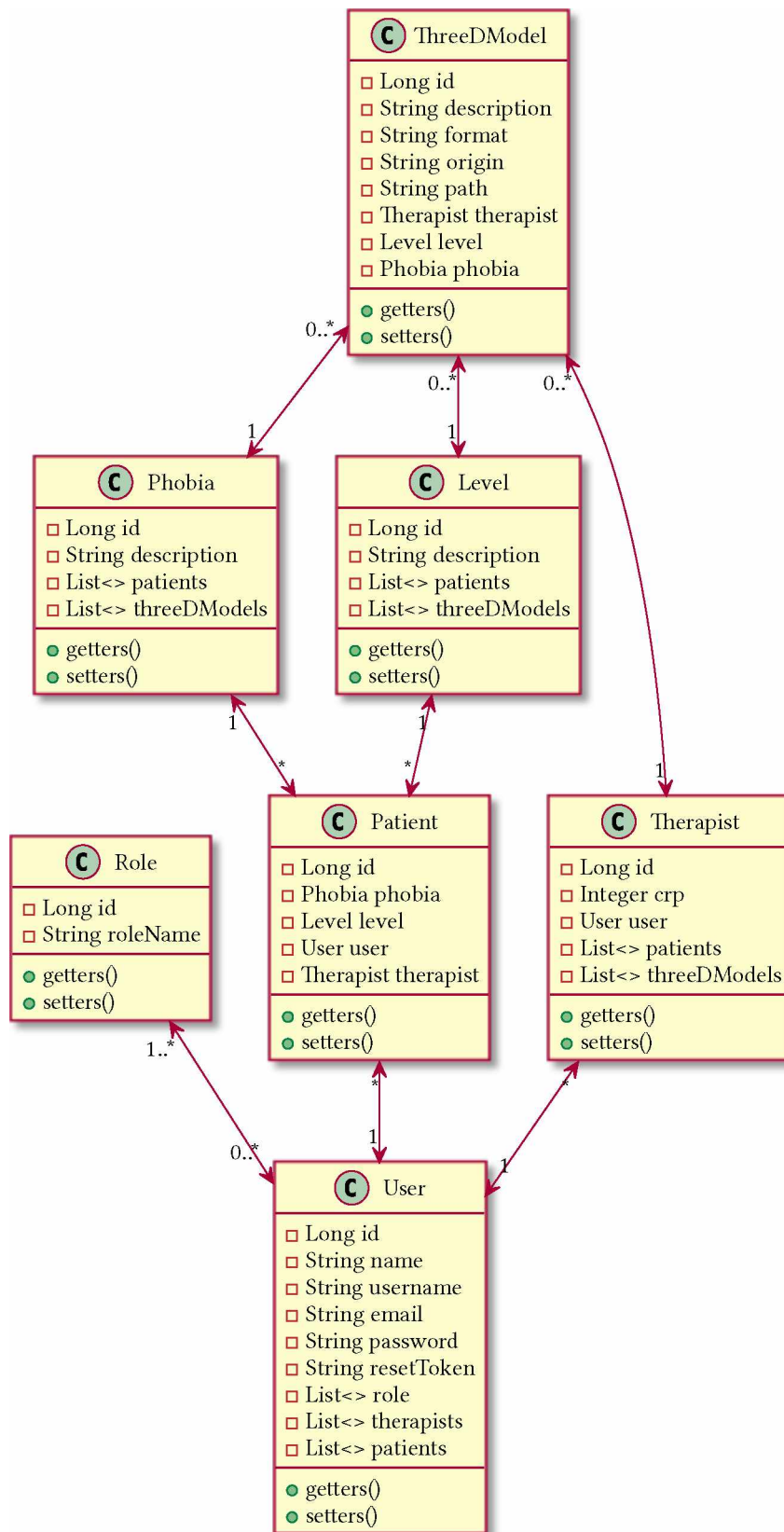
Esta implementação é possível a partir da utilização do Spring Data JPA. Esta ferramenta permite a utilização de anotações nas classes que realizam o mapeamento entre os atributos de cada entidade e as correspondentes tabelas no banco de dados.

Primeiramente, foram definidas as entidades necessárias à aplicação. São elas:

- *User*: entidade principal do usuário, contendo atributos comuns a terapeutas e pacientes;
- *Therapist*: entidade utilizada para representar os terapeutas. Contém campos específicos ao terapeuta, como a lista de pacientes e seus respectivos modelos 3D.
- *Patient*: entidade utilizada para representar os pacientes. Contém campos específicos aos pacientes como fobia e nível de tratamento;
- *Phobia*: entidade representando uma fobia específica;
- *Level*: entidade representando um nível de tratamento específico;
- *ThreeDModel*: entidade representando um modelo 3D. Através do nome do arquivo no banco de dados, o modelo 3D é mapeado com o sistema de arquivos. Cada modelo 3D possui uma fobia específica e um nível associado;
- *Role*: entidade representando o papel do usuário, que pode ser “ROLE_THERAPIST” ou “ROLE_PATIENT”. Este campo é utilizado para restringir o acesso de determinado usuário à determinadas áreas da aplicação de *front end* e *back end*;

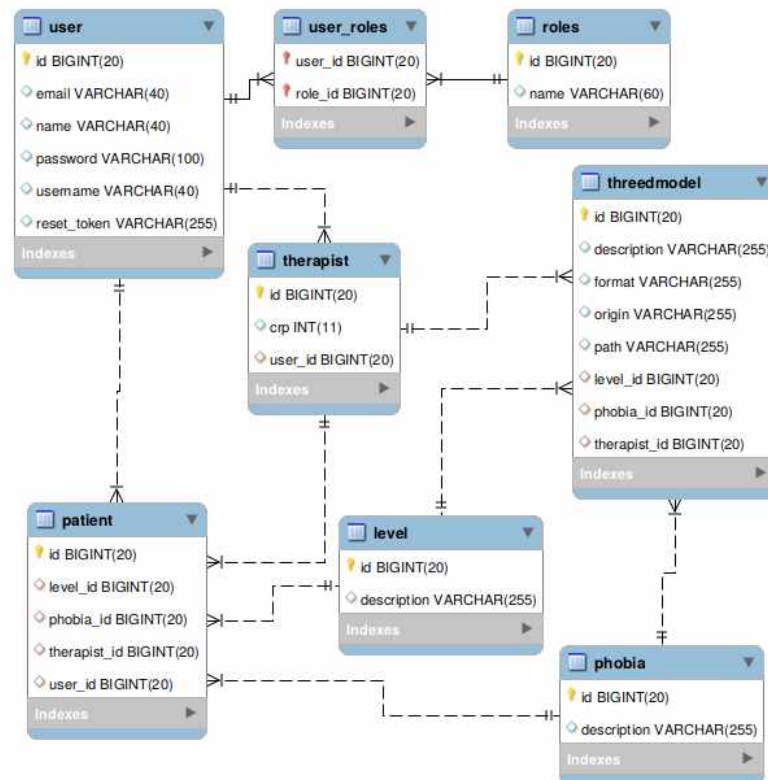
O diagrama representado na Figura 4.5 representa as tabelas e seus relacionamentos resultantes no banco de dados. Pode-se perceber onde são realizadas as conversões nos tipos de dados e a formação dos relacionamentos entre as entidades.

Figura 4.4: Diagrama de classes das entidades no *back end*.



Fonte: O autor (2019).

Figura 4.5: DER (Diagrama Entidade Relacionamento) do banco de dados.



Fonte: O autor (2019).

A partir definição das entidades, o Spring JPA possui a funcionalidade de geração e execução do código SQL equivalente no banco de dados, criando assim, as tabelas e implementando os seus relacionamentos.

A utilização de anotações no *Java* permite a redução de código *boilerplate*, dando ao programador a possibilidade de focar na estrutura e funcionalidade do código. Assim, o desenvolvedor se preocupa menos com questões de configuração e repete menos código durante o construção do sistema, focando nas regras de negócio do mesmo.

Como exemplo, a anotação `@Data` (linha 2) realiza a geração de todos os métodos *Getters* e *Setters* diretamente no *bytecode* da classe, baseando-se nos atributos declarados. A biblioteca responsável pela geração destes códigos é o *Lombok*¹⁴

Além disso, as anotações permitem que o comportamento da classe seja alterado de forma sucinta. Por exemplo, a anotação `@JsonIgnoreProperties` (linha 5) permite que determinados atributos sejam ignorados (não sejam retornados) nos *payloads* das requisições.

O trecho de código a seguir exhibe a implementação da classe *Patient*, destacando o mapea-

¹⁴Documentação disponível em: <https://projectlombok.org/>

mento do relacionamento das tabelas feito através das anotações `@ManyToOne` e `@JoinColumn`. Outras classes possuem a anotação `@OneToOne` e `@ManyToMany`.

```
1  @Entity
2  @Data
3  @NoArgsConstructor
4  @AllArgsConstructor
5  @JsonIgnoreProperties(value = {"id"})
6  public class Patient {
7
8      @Id
9      @GeneratedValue(strategy = GenerationType.IDENTITY)
10
11      private Long id;
12
13      @ManyToOne(fetch = FetchType.LAZY)
14      @JoinColumn(name = "phobia_id")
15      private Phobia phobia;
16
17      @ManyToOne(fetch = FetchType.LAZY)
18      @JoinColumn(name = "level_id")
19      private Level level;
20
21      @ManyToOne(fetch = FetchType.LAZY, cascade = CascadeType.ALL)
22      @JoinColumn(name = "user_id")
23      private User user;
24
25      @ManyToOne(fetch = FetchType.LAZY)
26      @JoinColumn(name = "therapist_id")
27      private Therapist therapist;
28  }
```

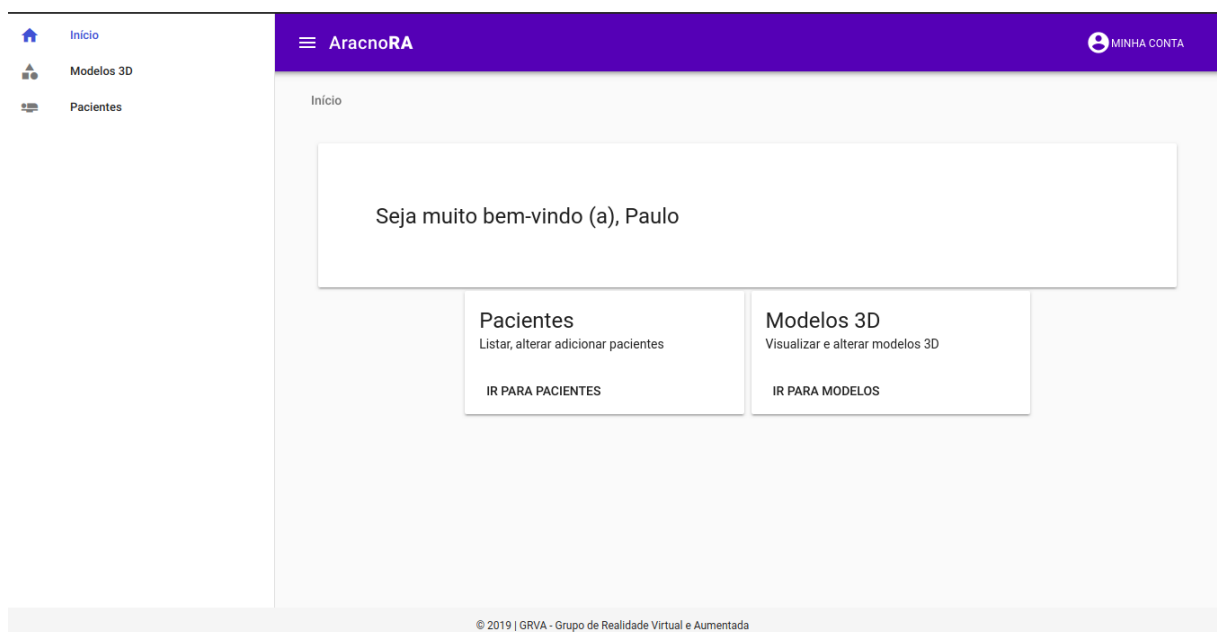

Capítulo 5

Resultados

5.1 Perfis do Sistema

O sistema possui dois tipos de perfis: um perfil para o terapeuta, com a tela inicial representada pela Figura 5.1 e um perfil para o paciente, com a tela inicial representada pela Figura 5.2. As figuras a seguir exibem as telas de ambos usuários:

Figura 5.1: Tela inicial do perfil do terapeuta.

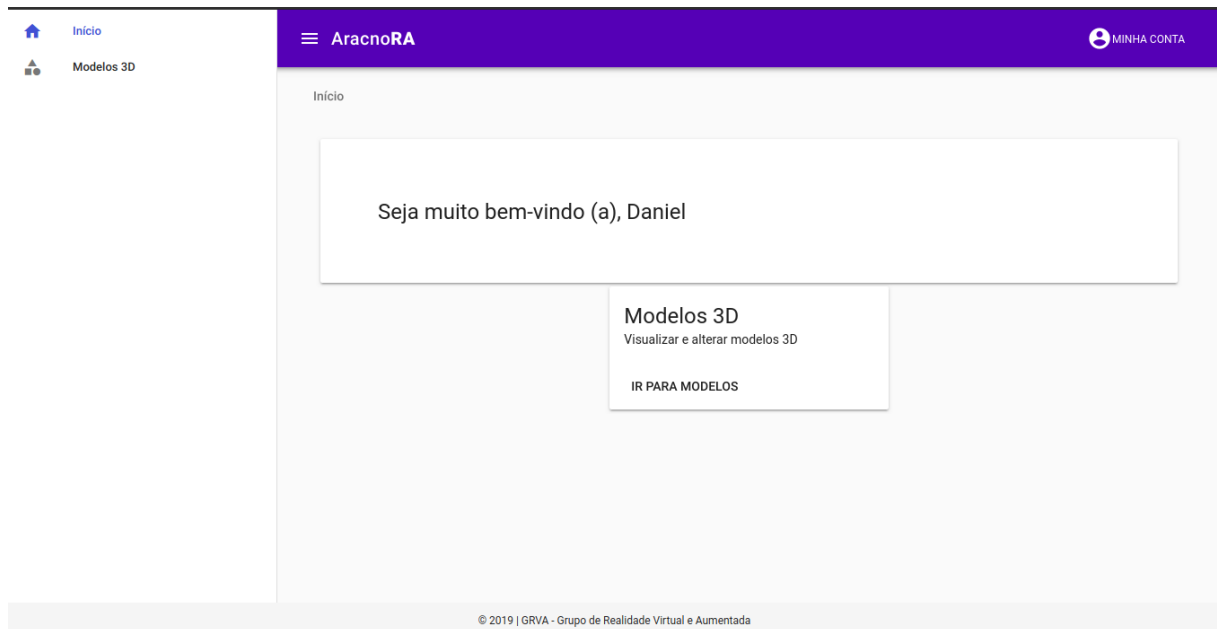


Fonte: O autor (2019).

A diferença entre os dois perfis é em relação a adição e edição de pacientes e modelos 3D. Enquanto os terapeutas podem adicionar, editar, remover e visualizar informações dos modelos 3D e pacientes, o perfil de paciente pode somente visualizar suas informações próprias e dos

modelos 3D liberados pelo seu terapeuta responsável.

Figura 5.2: Tela inicial do perfil do paciente.

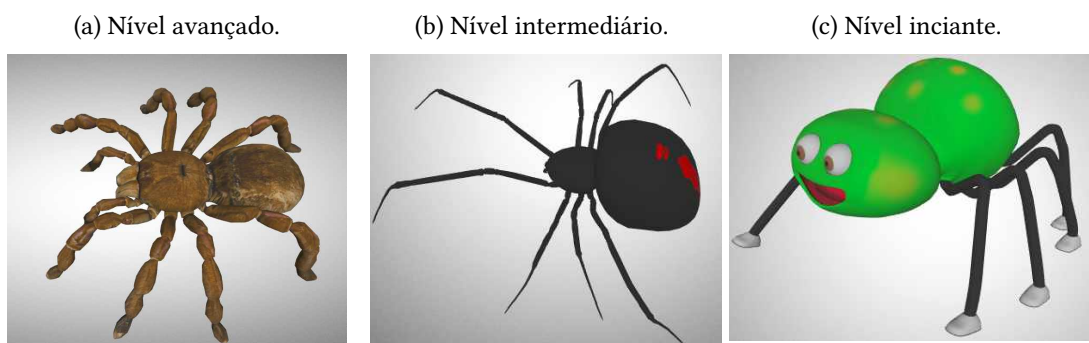


Fonte: O autor (2019).

5.2 Cadastro de Modelos 3D

O cadastro de modelos é feito por terapeutas e requer algumas informações básicas obrigatórias, como nível do modelo, fobia específica (aracnofobia pré-selecionada) e o arquivo do modelo, que deve ser em formato *glTF* (*GL Transmission Format*). A Figura 5.3 exibe exemplos de modelos 3D que podem ser adicionados no sistema.

Figura 5.3: Exemplos de modelos 3D que podem ser cadastrados no sistema.



Fonte: O autor (2019).

Este modelo 3D foi produzido especificamente para o sistema, porém qualquer outro que

respeite o formato *glTF* pode ser adicionado. Pela internet é possível encontrar *websites* específicos para *download* de modelos 3D animados e inanimados. A Figura 5.4 exibe a tela de cadastro de modelos 3D.

Figura 5.4: Tela de cadastro de modelos 3D.

A imagem mostra a interface web do sistema AracnoRA para o cadastro de um novo modelo 3D. No topo, há uma barra de navegação com o nome 'AracnoRA' e um link para 'MINHA CONTA'. À esquerda, um menu lateral contém as opções 'Início', 'Modelos 3D' (destacada) e 'Pacientes'. O conteúdo principal, sob o título 'Modelos 3D', apresenta o caminho de navegação 'Início > Modelos 3D > Novo modelo 3D'. Abaixo, o formulário 'Cadastrar novo modelo 3D' possui um campo de texto para 'Descrição', dois menus suspensos para 'Fobia' e 'Level', e uma seção para upload de arquivos com um ícone de mais (+) e o texto 'Arquivo Nenhum arquivo selecionado'. Um botão verde 'CADASTRAR MODELO 3D' está na base do formulário. O rodapé da página indica '© 2019 | GRVA - Grupo de Realidade Virtual e Aumentada'.

Fonte: O autor (2019).

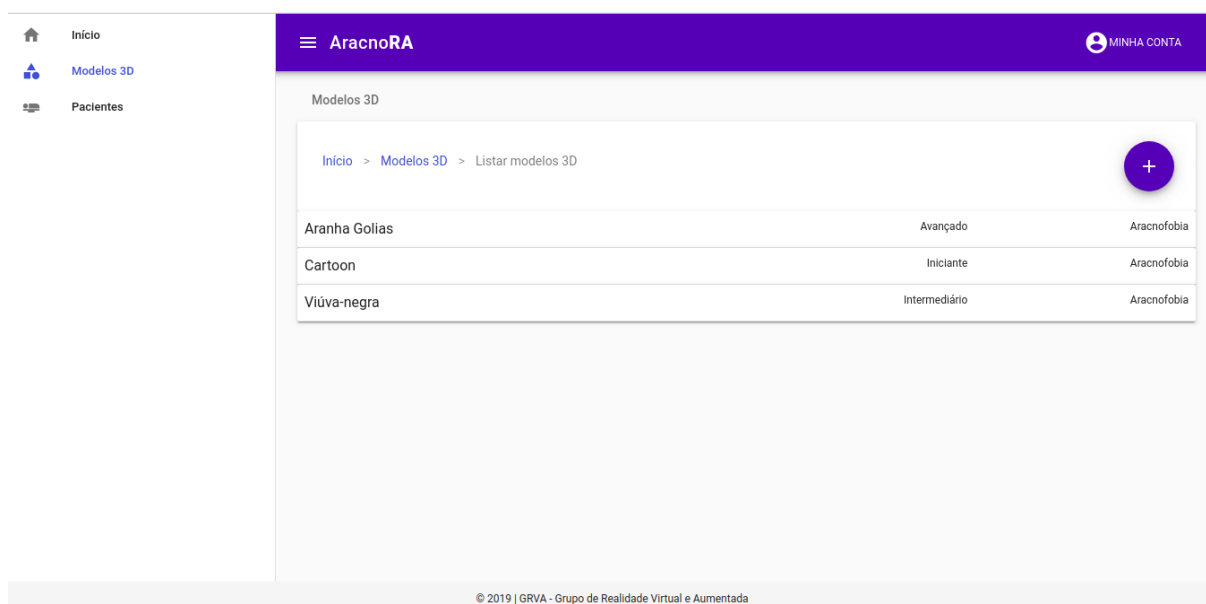
5.3 Visualização de Modelos 3D

A visualização do modelo 3D é feita através do próprio navegador do usuário. Ao clicar na opção “Ir para modelos” na tela inicial, o usuário pode selecionar a opção “Listar modelos”. No caso do terapeuta, a lista de todos os modelos cadastrados é apresentada.

A lista de modelos para o paciente exibe todos os modelos que o seu terapeuta responsável cadastrou e esteja abaixo do seu nível de tratamento atual, no caso de perfil do paciente. O diagrama para este caso pode ser observado na Figura 4.3.

No caso de perfil do terapeuta, a lista exibe todos os modelos que ele cadastrou no sistema. A lista permite que o usuário visualize algumas informações de antemão, como a fobia e o nível do modelo 3D. Um exemplo de lista de modelos 3D pode ser observado na Figura 5.5 a seguir.

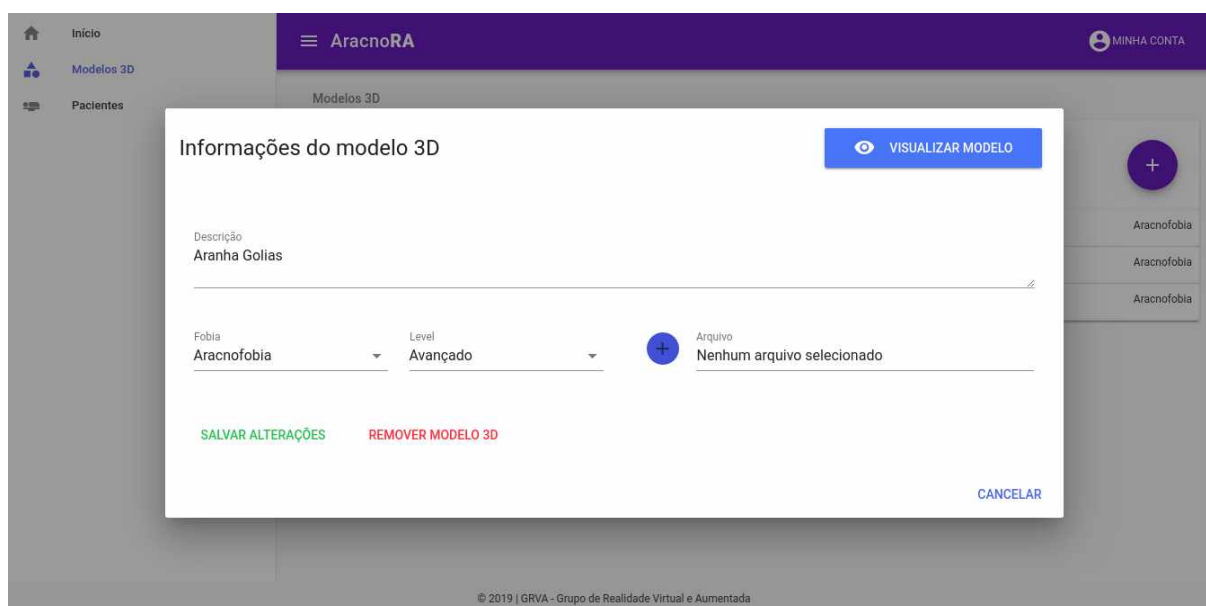
Figura 5.5: Lista de modelos 3D cadastrados.



Fonte: O autor (2019).

Ao clicar em um dos itens da lista, as informações do modelo são exibidas e o usuário pode clicar no botão “Visualizar modelo”. A Figura 5.6 exibe a tela para edição das informações de um modelo 3D. Nesta tela, o usuário pode selecionar a opção no botão “Visualizar”.

Figura 5.6: Tela de edição de informações de modelos 3D.



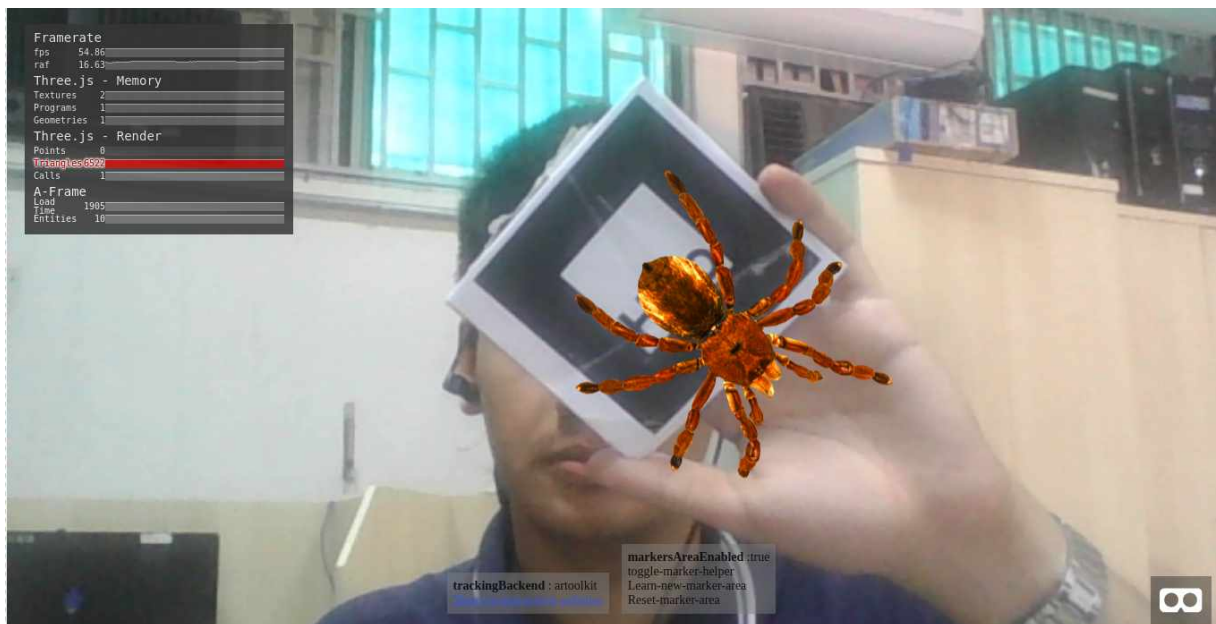
Fonte: O autor (2019).

Ao clicar em “Visualizar modelo” uma nova aba é aberta no navegador do usuário e o

modelo 3D é transferido do servidor para a máquina do usuário. Uma solicitação de acesso à *webcam* do usuário é feita, de modo a permitir a sua utilização.

Ao iniciar o vídeo (após aceitar a solicitação de acesso à *webcam*), o usuário deve posicionar o marcador de frente a tela, de forma que seja visível para a *webcam*. O modelo é então renderizado na posição do marcador. A animação do modelo também é iniciada, caso presente. A Figura 5.7 exibe um exemplo de utilização com um modelo de aranha sendo renderizado.

Figura 5.7: Exemplo de renderização de modelo 3D na tela do computador.



Fonte: O autor (2019).

Ao posicionar o marcador pela primeira vez de frente a tela, uma requisição é enviada ao servidor solicitando o arquivo do modelo 3D. O *download* é então iniciado o modelo é armazenado no *cache* do navegador. As exibições de marcador posteriores utilizam o modelo 3D armazenado no *cache*. Desta forma, a interação com o usuário se torna mais fluida.

A Figura 5.7 exibe um quadro no canto superior esquerdo contendo informações referentes ao modelo 3D carregado e sua renderização. A partir de testes efetuados em *notebooks* com diferentes configurações de *hardware*, foi possível observar que a taxa FPS (Frames per Second) do modelo de mais alto nível (maior realismo) inicia com valores próximos de 20 FPS durante o carregamento, porém, tende a estabilizar-se na faixa de 50 a 55 FPS, em todos os dispositivos.

Conclusões e Trabalhos Futuros

Este trabalho apresentou a construção de um sistema de tratamento de Aracnofobia utilizando Realidade Aumentada. Todo o sistema foi desenvolvido de forma a permitir a portabilidade. Desta forma, sua implantação é feita em um servidor web, onde qualquer computador e smartphone pode acessar.

A diminuição dos custos de tecnologias relacionadas à informática, bem como melhorias no processamento e armazenamento de computadores e *smartphones*, permitiram que estes dispositivos se tornassem comuns e de fácil acesso. Este trabalho aproveita estes recursos para fornecer uma plataforma compatível com estes dispositivos.

Por ser acessado através da internet, esta plataforma permite que o tratamento ocorra mesmo à distância. O terapeuta pode solicitar a um paciente a execução de um determinado exercício de exposição em sua própria residência. Esta possibilidade reduz a necessidade de deslocamentos de terapeutas e pacientes, comum nos tratamentos convencionais.

Além disso, com a técnica de RA, permite-se que sejam utilizadas aranhas virtuais. Esta característica elimina a necessidade de utilização de aranhas reais, que muitas vezes causam a desistência do tratamento por parte de pacientes. Este método dá ao terapeuta um maior controle sobre o comportamento do objeto no ambiente, tendo em vista que as animações dos modelos 3D já estão prontas.

A possibilidade de utilização de aranhas com níveis de realismo incremental, permite ao terapeuta a exposição gradual dos modelos aos pacientes, baseando-se no nível de evolução dos mesmos. O terapeuta pode adicionar à sua biblioteca quaisquer modelos que achar adequado.

O sistema ainda possui um potencial de ampliação para outras fobias de animais, como a cinofobia (fobia de cães), ofidiofobia (fobia de serpentes), murofobia (fobia de ratos), entre

outros. Para isto, seria necessário necessárias pequenas adaptações, como incluir um formulário para cadastro de fobias, no sistema para permitir a adição de modelos 3D destes animais.

Como trabalho futuro, pretende-se a realização de testes de exposição com pacientes, utilizando técnicas de Reconsolidação de Memória. Além disso, pretende-se retirar a necessidade da utilização de marcadores do sistema. Esta remoção de marcadores pode ser feita através de técnicas de geolocalização (GPS), por exemplo.

O sistema atualmente conta com técnica de RA não imersiva, com exposição em tela. Uma proposta a ser considerada é a utilização de óculos apropriados de RA como HoloLens[®] que permitem mecanismos de interação para que usuários possam visualizar os modelos e interagir com os mesmos através de gestos com a mão ou comandos de voz.

Bibliografia

APA, American Psychiatric Association -. **Diagnostic and statistical manual of mental disorders (DSM-5®)**. [S.l.]: American Psychiatric Pub, 2013.

BACCA, Jorge et al. Augmented reality trends in education: a systematic review of research and applications. National Sun Yat-sen University, 2014.

BARSOM, EZ; GRAAFLAND, M; SCHIJVEN, MP. Systematic review on the effectiveness of augmented reality applications in medical training. **Surgical endoscopy**, Springer, v. 30, n. 10, p. 4174–4183, 2016.

BERNHARDT, Sylvain et al. The status of augmented reality in laparoscopic surgery as of 2016. **Medical image analysis**, Elsevier, v. 37, p. 66–90, 2017.

BOETTCHER, Hannah; BRAKE, C Alex; BARLOW, David H. Origins and outlook of interoceptive exposure. **Journal of Behavior Therapy and Experimental Psychiatry**, Elsevier, v. 53, p. 41–51, 2016.

BOTELLA, Cristina; BAÑOS, Rosa M et al. Virtual reality in the treatment of claustrophobic fear: A controlled, multiple-baseline design. **Behavior therapy**, Elsevier, v. 31, n. 3, p. 583–595, 2000.

BOTELLA, Cristina; BRETÓN-LÓPEZ, Juani et al. Treating cockroach phobia with augmented reality. **Behavior Therapy**, Elsevier, v. 41, n. 3, p. 401–413, 2010.

BOTELLA, Cristina; FERNÁNDEZ-ÁLVAREZ, Javier et al. Recent progress in virtual reality exposure therapy for phobias: a systematic review. **Current psychiatry reports**, Springer, v. 19, n. 7, p. 42, 2017.

BOTELLA, Cristina; PÉREZ-ARA, M Ángeles et al. In vivo versus augmented reality exposure in the treatment of small animal phobia: a randomized controlled trial. **PloS one**, Public Library of Science, v. 11, n. 2, e0148237, 2016.

BOURDON, Karen H et al. Gender differences in phobias: results of the ECA community survey. **Journal of Anxiety Disorders**, Elsevier, v. 2, n. 3, p. 227–241, 1988.

CHICCHI GIGLIOLI, Irene Alice et al. Augmented reality: a brand new challenge for the assessment and treatment of psychological disorders. **Computational and mathematical methods in medicine**, Hindawi, v. 2015, 2015.

CHOY, Yajuan; FYER, Abby J; LIPSITZ, Josh D. Treatment of specific phobia in adults. **Clinical psychology review**, Elsevier, v. 27, n. 3, p. 266–286, 2007.

FIELDING, Roy T; TAYLOR, Richard N. **Architectural styles and the design of network-based software architectures**. [S.l.]: University of California, Irvine Irvine, USA, 2000. v. 7.

FOWLER, Martin; KOBRYN, Cris; SCOTT, Kendall. **UML distilled: a brief guide to the standard object modeling language**. [S.l.]: Addison-Wesley Professional, 2004.

JOSEPH, Renien John. Single page application and canvas drawing. **arXiv preprint arXiv:1502.03530**, 2015.

JUAN, M Carmen et al. Using augmented reality to treat phobias. **IEEE computer graphics and applications**, IEEE, v. 25, n. 6, p. 31–37, 2005.

KRIJN, Merel et al. Fear of flying treatment methods: virtual reality exposure vs. cognitive behavioral therapy. **Aviation, space, and environmental medicine**, Aerospace Medical Association, v. 78, n. 2, p. 121–128, 2007.

MAPLES-KELLER, Jessica L et al. Virtual reality-enhanced extinction of phobias and post-traumatic stress. **Neurotherapeutics**, Springer, v. 14, n. 3, p. 554–563, 2017.

MARKS, IM; O’SULLIVAN, G. Tratamiento de exposición en la agorafobia y el pánico. **Avances en el tratamiento psicológico de los trastornos de ansiedad**, Pirámide Madrid, p. 35–55, 1992.

MEDEIROS, Danielly Cristine et al. Realidade virtual não-imersiva como tecnologia de apoio no desenvolvimento de protótipos para o auxílio no tratamento de aviofobia por profissionais de psicologia. **Uberlândia: UFU**, 2004.

MESBAH, Ali; VAN DEURSEN, Arie. Migrating multi-page web applications to single-page Ajax interfaces. In: IEEE. 11TH European Conference on Software Maintenance and Reengineering (CSMR'07). [S.l.: s.n.], 2007. p. 181–190.

MILOFF, Alexander et al. Single-session gamified virtual reality exposure therapy for spider phobia vs. traditional exposure therapy: study protocol for a randomized controlled non-inferiority trial. **Trials**, BioMed Central, v. 17, n. 1, p. 60, 2016.

MISHKIND, Matthew C et al. Review of virtual reality treatment in psychiatry: evidence versus current diffusion and use. **Current psychiatry reports**, Springer, v. 19, n. 11, p. 80, 2017.

MORINA, Nexhmedin et al. Can virtual reality exposure therapy gains be generalized to real-life? A meta-analysis of studies applying behavioral assessments. **Behaviour research and therapy**, Elsevier, v. 74, p. 18–24, 2015.

SIELHORST, Tobias; FEUERSTEIN, Marco; NAVAB, Nassir. Advanced medical displays: A literature review of augmented reality. **Journal of Display Technology**, IEEE, v. 4, n. 4, p. 451–467, 2008.

SILBERSCHATZ, Abraham; SUNDARSHAN, S; KORTH, Henry F. **Sistema de banco de dados**. [S.l.]: Elsevier Brasil, 2016.

SUSO-RIBERA, Carlos et al. Virtual Reality, Augmented Reality, and In Vivo Exposure Therapy: A Preliminary Comparison of Treatment Efficacy in Small Animal Phobia. **Cyberpsychology, Behavior, and Social Networking**, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New ..., v. 22, n. 1, p. 31–38, 2018.

WRZESIEN, Maja et al. Treating small animal phobias using a projective-augmented reality system: A single-case study. **Computers in Human Behavior**, Elsevier, v. 49, p. 343–353, 2015.